# <Online Drugstore System>

## Object Design

## <1.0>

## <29.12.2017>

<EMRE YILMAZ
ÇAĞLA ÇINAR
ERTUĞRUL KOÇ
SEFA KORKMAZ>

Prepared for
SE301 Software Engineering

IŞIK UNIVERSITY
COMPUTER
SCIENCE AND
ENGINEERING

# Table of Contents

# OBJECT DESIGN DOCUMENT

## 1. Introduction

This Object Design Document defines the design of an online shopping system.The goal of this project is provide a reliable service for people and its main purpose simplify buying an online drug. Also it provides help for doctors for use online drugstore system for selling their drug in online.

The Online Drugstore System has trade off which leads to some costs like as: development, understand ability and security.

### 1.1. Object Design Trade-offs

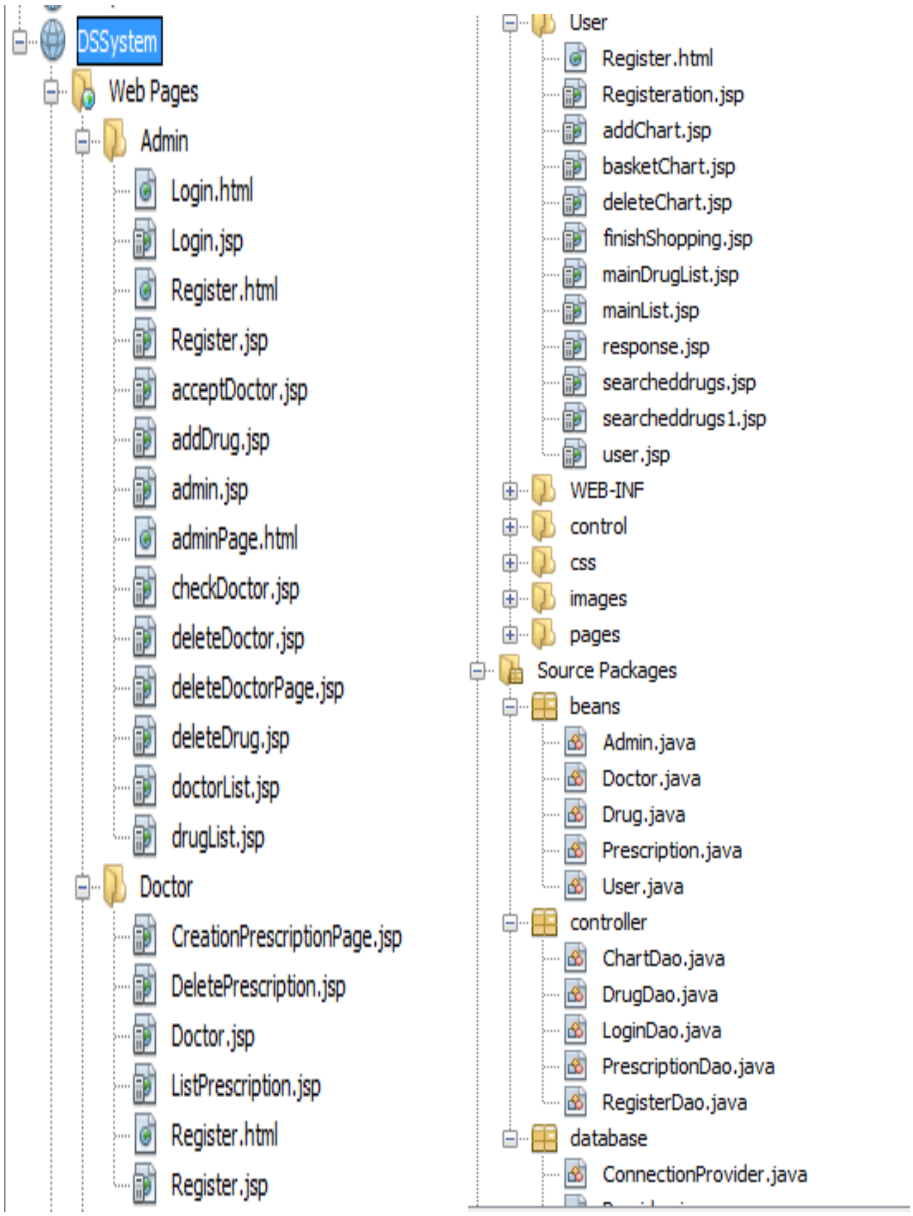
Object Design trade-offs leads to some costs like as: development,understand ability and security.

Online Drugstore System users, doctors and admin use functions of the system for using it.That causes development cost because all that functions takes time for design and implement.

When developers implement the code of the system it needs to be understand ability because when they need to change something missing or wrong they can’t lose time for found where that excuse.They need to use commend lines for avoid that time lose.

In Online Drugstore System all users needs to be registered.Unregistered users cannot use the system.Also doctors when they registered they need to wait admin approval.All of guests need an email and password.This situation requires an security cost.

## 1.2. Interface Documentation Guidelines




Picture 1.1- Drugstore System Packages And Classes

| Fully qualified class name | Unqualified name | Remarks |
|---|---|---|
| **login** | JSP File | Admin,Doctor and User's login class |
| **register** | JSP File | Doctor and User's Register class |
| **acceptDoctor** | JSP File | Admin accept request in this class |
| **addDrug** | JSP File | Admin's adding drug class |
| **admin** | JSP File | Admin main page |
| **checkDoctor** | JSP File | Admin's control the doctors's request class |
| **deleteDoctor** | JSP File | Admin's delete doctor class |
| **deleteDrug** | JSP File | Admin's delete drug class |
| **doctorList** | JSP File | Admin's list of doctors class |

| | | |
|---|---|---|
| **drugList** | JSP File | Admin's list of drug class |
| **CreationPrescriptionPage** | JSP File | Doctor's adding prescription class |
| **DeletePrescription** | JSP File | Doctor's delete prescription class |
| **Doctor** | JSP File | Doctor main page |
| **ListPrescription** | JSP File | Doctor's list of prescription class |
| **Register** | JSP File | User and Doctor's register class |
| **addChart** | JSP File | User's adding chart class |
| **deleteChart** | JSP File | User's deleting chart class |
| **basketChart** | JSP File | Chart page |
| **finishShopping** | JSP File | User's finish shopping class |
| **mainDrugList** | JSP File | User's drugList for main page class |
| **response** | JSP File | Credit card response class |
| **searchedDrugs** | JSP File | Search with DrugList class |
| **user** | JSP File | User main page |
| | | |
| **Login** | HTML File | User,Admin,Doctor Login form |
| **Register** | HTML File | User,Admin,Doctor Register form |
| | | |
| **Admin** | JAVA File | The class for which the admin was created |
| **Drug** | JAVA File | The class for which the drug was created |
| **Doctor** | JAVA File | The class for which the doctor was created |
| **Prescription** | JAVA File | The class for which the prescription was created |
| **User** | JAVA File | The class for which the user was created |
| **ChartDao** | JAVA File | Class with methods related to chart |
| **DrugDao** | JAVA File | Class with methods related to drug |
| **DoctorDao** | JAVA File | Class with methods related to doctor |
| **LoginDao** | JAVA File | Class with methods related to login |
| **RegisterDao** | JAVA File | Class with methods related to register |
| **ConnectionProvider** | JAVA File | Connection class with database |
| **Provider** | JAVA File | The class in which Mysql database's information is kept |

### 1.3. Definitions, Acronyms, and Abbreviations

- Database: Database is a collection of information that is organized so that it can easily be Accessed

-MYSQL: My Structured Query Language

- ID : Identification

- Login: To get access to an operating system or application, usually in a remote computer

- ODD : Object Design Document

- UI : User Interface

- JSP: Java Server Pages

- HTML : HyperText Markup Interface

-CSS: Cascading Style Sheets

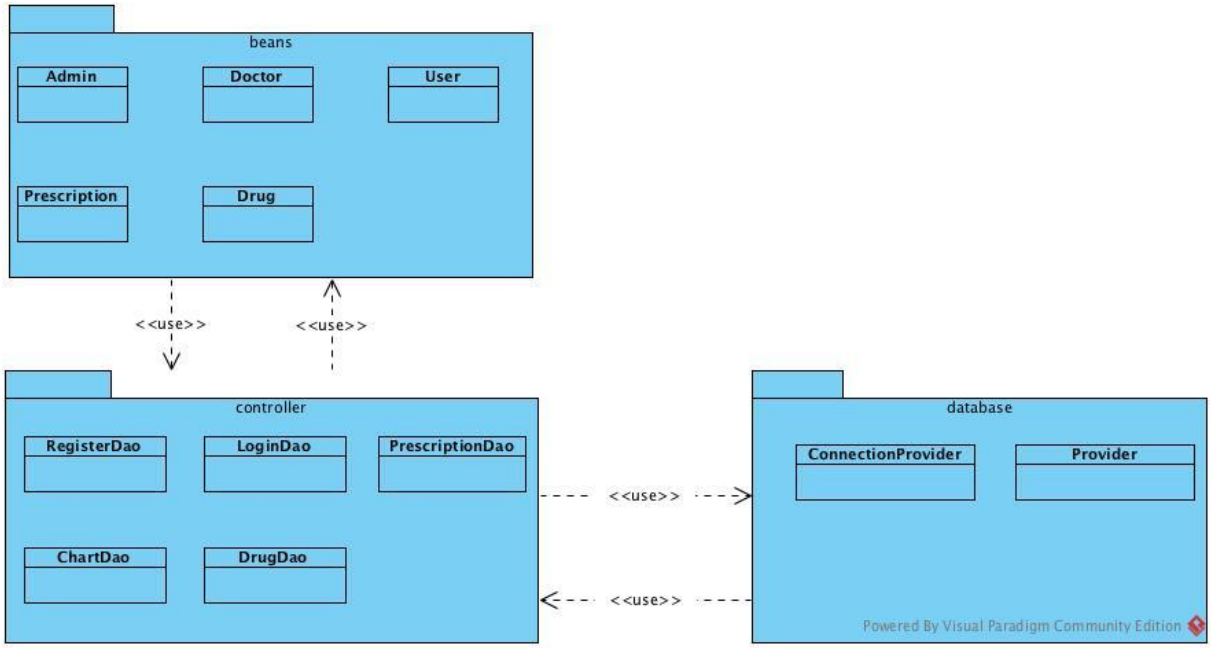
-Session: Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc). By default, session variables last until the user closes the browser.

- Exception: Represents errors that occur during application execution.

### 1.4. References

http://blog.slickedit.com/2007/05/how-to-write-an-effective-design-document/
http://ee.hawaii.edu/~tep/EE467/BDLecs/html/odd.htm

## 2. Packages




First of all, we defined variables in certain classes in the beans package. This allows you to exchange data between the target database and the control. For example, when the user in the User class is pulling from the database, the username in the user class in the beans matches the desired username and allows you to receive data from the database class. Another package is database. The database is the layer that provides the communication between the beans and the control package. The database package provides our connection with our existing database. And these packages we can make database operations easy. It contains our basic functions in the class of controls. The controls package includes of classes that contain the basic functions. For example, we need a listing method to list user information. When we call this method from where we want it, we can list our data.

## 3. Class Interfaces

As shown in the picture above *(Picture1.1)*, Online Drugstore System consists of different classes and packages. Java classes have the database connection, methods and attributes. JSP classes help to call the methods and create the web pages' interface.DS System use the HTML classes for create login, register and finish shopping form.

**Java Classes in database Packages**

**Provider.java:** This class has to attribute for the database connection.

**ConnectionProvider.java:** This class helps us to connect to database. This class takes the database login information from Provider.java class.

**Java Classes in controller Packages**

Every class in this package connects to ConnectionProvider.java for using the database.

**ChartDao.java:** This class includes the methods for the chart. These methods are;

| Methods | Method Task |
|---|---|
| **public static void createBasket(beans.Drug d, String useremail)** | Thanks to this method, users add to drug to chart when buying. |
| **public static ResultSet takeAllBasket(String email) throws SQLException** | Thanks to this method, users see all drugs taken in chart. |
| **public static boolean deleteBasket(String name) throws SQLException** | Thanks to this method, users delete the selected drug from chart. |
| **public static boolean finishShopping(String firstname) throws SQLException** | Thanks to this method, system takes the user information in database and finishes the shopping. |

These methods are used by the some JSP classes in the User package.

**DrugDao.java:** This class includes the methods for the drug. These methods are;

| Methods | Method Task |
|---|---|
| **public static void addDrug(beans.Drug d)** | Thanks to this method, admin add to drug database |
| **public static ResultSet takeAllDrug() throws SQLException {** | Thanks to this method, admin can see all drugs in the database |
| **public static boolean deleteDrug(String drugname) throws SQLException** | Thanks to this method, admin can delete the selected drug from drug list. |
| **public static void updateDrug(String drugname) throws SQLException** | Thanks to this method, admin update the drug information in database. |

These methods are used by the some JSP classes in the Admin package.

**PrescriptionDao.java:** This class includes the methods for the prescription. These methods are;

| Methods | Method Task |
|---|---|
| **Public static void create Prescription (beans.Prescription d) {** | Thanks to this method, doctors add to prescription database |
| **public static ResultSet takeAllPrescription() throws SQLException** | Thanks to this method, doctors can see all prescription in the database |
| **public static boolean delete Prescription (String doctorName) throws SQLException** | Thanks to this method, doctors can delete the selected prescription from prescription list. |
| **public static void updateProfile() throws SQLException** | Thanks to this method, doctors update their information in database. |

These methods are used by the some JSP classes in the Doctor package.

**LoginDao.java:** This class includes the methods for the all actors. These methods are;

| Methods | Method Task |
|---|---|
| **public static boolean loginAdmin(String email, String password) throws SQLException** | Thanks to this method, system controls the admin information. |
| **public boolean checkDoctor(String email, String password) throws SQLException {** | Thanks to this method, system controls the doctor information. |
| **public boolean checkUser(String email, String password) throws SQLException** | Thanks to this method, system controls the user information. |
| **public static ResultSet checkStatu() throws SQLException** | Thanks to this method, admin control the request of doctor. |

These methods are used by the some JSP classes in the Doctor, Admin and User package.

**RegisterDao.java:** This class includes the methods for the all actors. These methods are;

| Methods | Method Task |
|---|---|
| **public static boolean registerUser(beans.User u)** | Thanks to this method, user information save the database |
| **Public static boolean registerDoctor (beans.Doctor d)** | Thanks to this method, doctor information save the database |
| **public static boolean registerAdmin (beans.Admin a)** | Thanks to this method, admin information save the database |
| **public boolean updateDoctorStatus(String demail, String status)** | Thanks to this method, Admin can change the statu of doctor. |

These methods are used by the some JSP classes in the Doctor, Admin and User package.

## Java Classes in beans Packages

**Admin.java:** This class includes the admin's attributes and these attributes have Getter, Setter methods. These methods are used by the some JSP classes in Admin package.

**Doctor.java:** This class includes the doctor's attributes and these attributes have Getter, Setter methods. These methods are used by the some JSP classes in Doctor package.

**User.java:** This class includes the user's attribute and these attributes have Getter, Setter methods. These methods are used by the some JSP classes in User package.

**Drug.java:** This class includes the drug's attributes and these attributes have Getter, Setter methods. These methods are used by the some JSP classes in Doctor, Admin and User package.

**Prescription.java:** This class includes the prescription's attributes and these attribute have Getter, Setter methods. These methods are used by the some JSP classes in Doctor, Admin and User package.

## JSP Classes

Online Drugstore System has many JSP classes. The task of these classes in DS system, create the interface and call the necessary methods. These classes use the HTML code for interface. If we want to call the one method, we should reference the package and class as in the fallowing examples. On the other hand JSP classes do not have the any public attributes in DS system.

Two sample JSP classes are shown below.

**Admin.jsp**

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<jsp:useBean id = "dbbean" scope = "session" class = "controller.LoginDao" >
</jsp:useBean>
<jsp:setProperty name = "dbbean" property = "*" />

<!DOCTYPE HTML>
<html>
   <head>
      <meta charset="UTF-8">
      <title>Drug Store</title>
      <link rel="stylesheet" type="text/css" href="../css/style.css" />
   </head>
   <body>
```

```
<%
   if (session.getAttribute("email") == null) {
      response.sendRedirect("../control/Login.jsp");
   }
   String email = (String) session.getAttribute("email");
   String firstname = dbbean.takeAdminName(email);
%>
<div id="header">
   <div class="clearfix">
      <div class="logo">
         <a href="index.html" title="fff"></a>
      </div>
      <ul class="navigation">
         <li class="active">
            <a href="admin.jsp">Admin: <%= firstname%></a>
         </li>

         <li>
            <a href="acceptDoctor.jsp">Accept Doctor</a>
         </li>

         <li>
            <a href="../control/Logout.jsp">Logout</a>
         </li>
      </ul>
   </div>
</div>
<div id="contents">
   <div class="clearfix">
      <div class="main">
         <h1>Admin Main Page</h1>

         <form action="addDrug.jsp" method="post" class="message">
            <br> <input type="submit" value="Add New Drug"></br>

         </form>

         <form action="drugList.jsp" method="post" class="message">
            <br> <input type="submit" value="Drugs List"></br>

         </form>

         <form action="doctorList.jsp" method="post" class="message">
            <br> <input type="submit" value="Doctors List"></br>
```

```
                </form>

            </div>
          </div>
        </div>
      </body>
    </html>
```

**Doctor.jsp**

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<jsp:useBean id = "dbbean" scope = "session" class = "controller.LoginDao" >
</jsp:useBean>
<jsp:setProperty name = "dbbean" property = "*" />

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Drug Store</title>
        <link rel="stylesheet" href="css/style.css" type="text/css">
        <link rel="stylesheet" type="text/css"
href="${pageContext.request.contextPath}/css/style.css" />
    </head>
    <body>
        <%
            if (session.getAttribute("email") == null) {
                response.sendRedirect("../control/Login.jsp");
            }
            String email = (String) session.getAttribute("email");
            String firstname = dbbean.takeDoctorName(email);
        %>

        <div id="header">
            <div class="clearfix">
                <div class="logo">

                </div>
                <ul class="navigation">
                    <li class="active">
                        <a href="Doctor.jsp">Doctor: <%= firstname%></a>

                    </li>
```

```
                <li>
                    <a href="../control/Logout.jsp">Logout</a>
                </li>
            </ul>
        </div>
    </div>

    <div id="contents">
        <div class="clearfix">
            <div class="main">
                <h1>Doctor Main Page</h1>

                <form action="CreationPrescriptionPage.jsp" method="post"
class="message">
                    <<br> <input type="submit" value="Create Prescription"></br>

                </form>

                <form action="CreationPrescriptionPage.jsp" method="post"
class="message">
                    <p><br> <input type="submit" value="add Prescription"></br><p>

                </form>

                <form action="ListPrescription.jsp" method="post" class="message">
                    <br> <input type="submit" value="Prescription List"></br>

                </form>

            </div>
        </div>
    </div>

    </body>
</html>
```